BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

# BACHELOR'S THESIS

Brno, 2021 Roman Santa


# BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

## DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ


# AUTOMATIC SPEECH RECORDINGS SEGMENTATION TOOL

NÁSTROJ PRO AUTOMATICKOU SEGMENTACI NAHRÁVEK ŘEČI

**BACHELOR'S THESIS**
BAKALÁŘSKÁ PRÁCE


**AUTHOR** — **Roman Santa**
AUTOR PRÁCE

**SUPERVISOR** — **Ing. Daniel Kováč**
VEDOUCÍ PRÁCE



**BRNO 2021**

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

# Bachelor's Thesis

Bachelor's study program **Telecommunication and Information Systems**

Department of Telecommunications

***Student:*** Roman Santa ***ID:*** 187421

***Year of study:*** 3 ***Academic year:*** 2020/21

**TITLE OF THESIS:**

## Automatic speech recordings segmentation tool

**INSTRUCTION:**

The main objective of the thesis is to implement the tool in Python programming language. It will automatically create annotation data for speech recording, according to which the segmentation in the WaveSurfer editing program is performing. To detect the beginning and the end of the spoken word, your voice activity detector will be programmed and tested (eg based on speech energy, etc.). It will be further and compared with a detector based on Google WebRTC Voice Activity Detection. In the next step, the recognition of various sentences, words, etc. will take place, for which the method of dynamic time warping will be used. The tool will expect a WAV recording at its input and a configuration file (at discretion). It will export an annotation file in WaveSurfer format on its output.

**RECOMMENDED LITERATURE:**

[1] Google WebRTC Voice Activity Detection (VAD) module [online]. [cit. 2020-09-09]. Dostupné z: https://www.mathworks.com/matlabcentral/fileexchange/78895-google-webrtc-voice-activity-detection-vad-module

[2] H.MANSOUR, Abdelmajid, Gafar ZEN ALABDEEN SALH a Khalid A. MOHAMMED. Voice Recognition using Dynamic Time Warping and Mel-Frequency Cepstral Coefficients Algorithms. International Journal of Computer Applications. 2015, 116(2), 34-41. DOI: 10.5120/20312-2362. ISSN 09758887. Dostupné také z: http://research.ijcaonline.org/volume116/number2/pxc3902362.pdf

***Date of project specification:*** 1.2.2021 ***Deadline for submission:*** 31.5.2021

***Supervisor:*** Ing. Daniel Kováč

**prof. Ing. Jiří Mišurec, CSc.**
Chair of study program board

## ABSTRACT

Automatic Segmentation tool processes recordings in order to extract voiced parts. It is important for further speech analysis to work only with extracted speech and not noise. For analysis of the di□erence between syllables of patients with parkinson disease and heatlhy ones, this segmentation tool should help with processing recordings. Goal of this thesis is to implement and test voice detectors with Google WebRTC detector and pick the best speech detector with minimal error rate. Also, develop a segmentation tool for given recordings and test voice recognition with dymanic time warping. Database from the Brain Diseases Analysis Laboratory was used. It contains czech and hungarian recordings with equal number of male and female as well as heathy and diseased patients. Energy detector performed as the best detetor in the tests. There was no significant di□erence in error rates between male and female or healthy and diseased patients. Recordings with lower Signal-to-Noise ratio were harder to process with an error rate starting at 12%. Based on the results, new detector for the segmentation tool was proposed to process examined recordings. Finally, dynamic time warping algorithm was tested with mel frequency cepstral coe□cients to recognize similarities between speakers.

## KEYWORDS

Speech Recognition, Segmentation, Voice Activity Detection, Dynamic Time Warping, Python

# Author's Declaration

| | |
|---|---|
| **Author:** | Roman Santa |
| **Author's ID:** | 187421 |
| **Paper type:** | Bachelor's Thesis |
| **Academic year:** | 2020/21 |
| **Topic:** | Automatic speech recordings segmentation tool |

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno .....

.....
author's signature*

*The author signs only in the printed version.

## ACKNOWLEDGEMENT

I would like to thank the advisor of my thesis, Ing. Daniel Kováč for his valuable comments and guidance throughout creation of this paper.

# Contents

**Introduction** **10**

**1 Speech signal pre-processing** **11**
1.1 Signal Sampling . . . . . 11
1.2 Pre-emphasis filtering . . . . . 11
1.3 Segmentation . . . . . 12

**2 Voice Activity Detectors** **13**
2.1 Short-Term Energy Detector . . . . . 14
2.2 Volume Detector with Zero-Crossing Rate . . . . . 15
2.3 Likelihood Ratio Test . . . . . 15
2.4 Mel-Frequency Cepstral Coe□cients . . . . . 16
2.5 Google WebRTC Voice Activity Detection . . . . . 18

**3 Speech Recognition** **19**
3.1 Dynamic Time Warping . . . . . 20
3.1.1 Using Cepstral coe□cients . . . . . 20

**4 Database** **21**

**5 Python implementation** **22**
5.1 Used libraries . . . . . 22
5.2 Voice activity detector structure . . . . . 22
5.3 Dynamic time warping application . . . . . 24

**6 Testing** **25**
6.1 Segmentation . . . . . 25
6.2 Receiver Operating Characteristic analysis . . . . . 26
6.3 Errors in detectors . . . . . 27

**7 Results** **28**
7.1 Voice Activity Detectors . . . . . 28
7.2 Dynamic Time Warping . . . . . 31

**8 Discussion** **33**
8.1 Voice Activity Detection for task 7 . . . . . 33
8.2 Task detection and dynamic time warping . . . . . 33

**Conclusion** **34**

**Bibliography** **35**

**Symbols and abbreviations** **37**

**A Content of the electronic attachment** **38**

# List of Figures


1.1 Frequency response of a filter with $\alpha$=0.95 . . . . . 11
1.2 Time series of a signal showing segments and overlapping (yellow stripes) . . . . . 12
2.1 Energy E[S] of each segment with displayed threshold (green) . . . . 14
2.2 Behavior of volume detector and zero-crossing rate . . . . . 15
2.3 Likelihood ratio performance preview . . . . . 16
2.4 Cepstral coe□cients of syllables /pa/ – /ta/ – /ka/ . . . . . 17
2.5 MFCC speech/noise distance preview . . . . . 17
3.1 Di□erence between syllables /a/ /i/ and /u/ . . . . . 19
3.2 Coe□cients of /a/, /i/ and /u/ vowel . . . . . 20
5.1 Cut o□ end of task 1 and task 2 detection . . . . . 23
6.1 Manual testing in wavesurfer . . . . . 25
6.2 Wavesurfer interface . . . . . 25
6.3 ROC Curve of Energy VAD . . . . . 26

# List of Tables


4.1 Speech tasks . . . . . 21
6.1 Error type table . . . . . 27
7.1 Detector quality for Czech and Hungarian recordings . . . . . 28
7.2 Healthy Control CZ Male . . . . . 29
7.3 Parkinson Disease CZ Male . . . . . 29
7.4 Healthy Control CZ Female . . . . . 29
7.5 Parkinson Disease CZ Female . . . . . 29
7.6 Healthy Control HU Male . . . . . 30
7.7 Parkinson Disease HU Male . . . . . 30
7.8 Healthy Control HU Female . . . . . 30
7.9 Parkinson Disease HU Female . . . . . 30
7.10 Syllable /a/ /i/ /u/ matching results . . . . . 31
7.11 Tested required distance for syllable /a/ match . . . . . 32
7.12 Tested required distance for syllable /i/ match . . . . . 32
7.13 Tested required distance for syllable /u/ match . . . . . 32

# Introduction

Human voice has become a powerful source of information in many science departments. Alongside informatics, voice analysis has advanced rapidly during last decade. A lot of new algorithms have been introduced by engineers around the world. From Short-Term Energy algorithm used for voice detection to advanced neural networks extracting various data for further analysis (e.g. emotion recognition, disease detection).

Voice Activity Detection (VAD) is a necessity in speech analysis. It detects beginning and the end of voiced parts ignoring unvoiced and noisy samples. Signal to noise ratio (SNR) plays crucial role in VAD. Heavy noised samples are hard to analyze and advanced techniques are needed to satisfy further requirements. Therefore, good microphone quality and quiet room lead to higher SNR and more satisfying results. There are plenty of VAD methods taking different approach. One of the fastest and modern nowadays is Google WebRTC VAD used for real-time communications. Short-term energy VAD is one of the most trivial but delivering good results when adapted to a specific task. Energy is a common property extracted from speech samples used in VAD. In speech recognition, our goal is to extract voice samples from sound clips as precise as possible to avoid unnecessary error in further analysis. Detector should take recording with other parameters as input and returns text file with .lab extension for further analysis with Wavesurfer application. Output file should contain time stamps in seconds of beginning and the end of spoken parts.

The purpose of this paper is to compare various detectors and select the most effective to build a segmentation tool for the recordings. This tool is supposed to break down recordings to specified speech tasks – from continuous speech to one breath word repetition. Output of the segmentation tool is a file with timestamps for further use and analysis.

This paper describes how speech pre-processing works as a first step before specific VAD method application. Including sampling, filtering and segmentation. Also, how detectors work and what types of detectors are used for testing. Google's detector is examined and briefly explained. Implementation of detectors is described, as well as used libraries and methodology. Testing was performed for specified database as described later in this paper. Further speech recognition was performed with Dynamic Time Warping showing room for improvement. Results were analyzed with manually segmented recordings and errors were computed.

# 1 Speech signal pre-processing

Continuous-time signal is processed to computer-like form as discrete-time series. Pre-processing tools are used to improve the results of segmentation and further analysis.

## 1.1 Signal Sampling

Real-world signals such as speech or audio signal in general are continuous. A process of converting continuous-time signal to a discrete sequence of numbers is called **sampling**. It is a process of picking values of a signal at specific time intervals depending on sampling frequency. **Sampling frequency** used for audio signals is often 44 100 Hz but for speech signals, 16 kHz is sufficient enough. Where most speech samples are bellow 3 kHz, giving speech samples space up to 8 kHz by the **Sampling theorem** [1].

## 1.2 Pre-emphasis filtering

In speech processing, many analysis methods tend to process a signal based on their high intensity spectrum. In general, most of the speech energy is bellow 1 kHz so in order not to lost information on higher frequencies pre-emphasis filter comes in hand. **Pre-emphasis** is a high-pass FIR filter used to flatten the spectrum of a signal [2]. It is often used as a first step in speech analysis. Pre-emphasis raises speech energy by a variable amount increased with frequency. Transfer function of pre-emphasis filter:

$$H(z) = 1 - \alpha z^{-1}, \tag{1.1}$$

where $\alpha$ is a coefficient usually between 0.9 and 1.

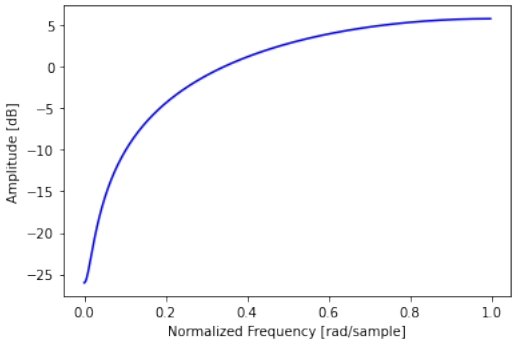


Fig. 1.1: Frequency response of a filter with $\alpha$=0.95

Pre-emphasis is an **optional** pre-processing step because it can affect the result

by enhancing high frequency noise. This noise can be reduced by good conditions during recording (e.g. microphone quality, silence room, etc.).

## 1.3 Segmentation

After sampling, there is a frequency rate, which the signal is sampled with and signal values as the result of previous step. In speech analysis, signal is often split into **segments** of defined size to evaluate behavior over short period of time (Fig. 1.2). Each spoken vowel has its characteristic properties such as length and frequency. **Length** of each segment is set to 10-30ms as the average vowel length in speech is in the interval. In some cases, the segment will include signal from middle of the vowel duration, so when later analyzed, the information about the whole vowel is lost. To prevent this scenario, **overlap** is used. Segments overlap up to 50% of their size. Each segment now contains 25% of the end of previous and 25% of the beginning of the next segment. There is almost double the data used in computing using 50% overlap, so it is important to select overlap with caution [3].

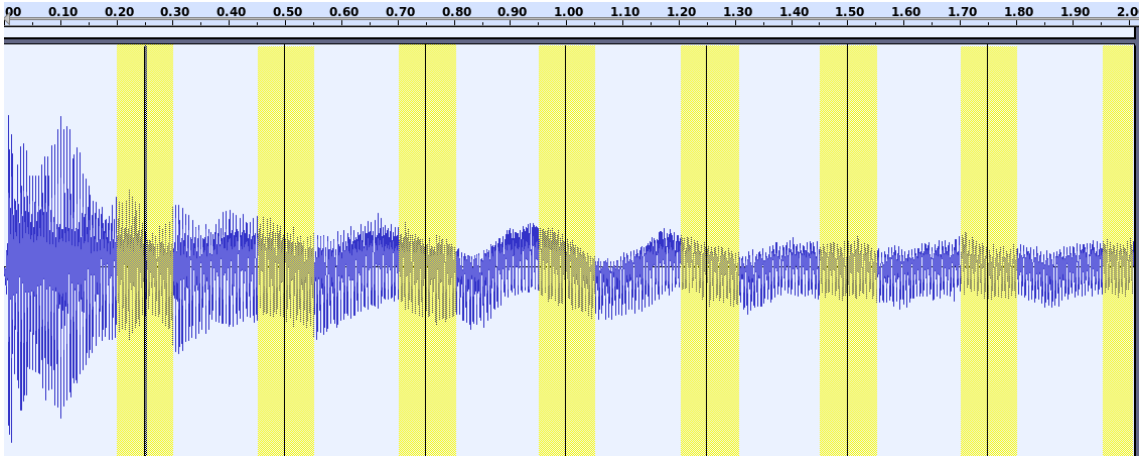


Fig. 1.2: Time series of a signal showing segments and overlapping (yellow stripes)

To amplify original (non-overlapped) segment data, window function is applied. **Hamming window**:

$$H(\theta) = 0.54 + 0.46 \cdot cos[(2\pi N)n], \tag{1.2}$$

have sinusoidal shape. Hamming window is characteristic by not touching zero value at both ends and its good results in DTFT (Discrete-Time Fourier Transform). After multiplying every value of each segment with corresponding window value, output contains respective values to their importance in further analysis.

# 2 Voice Activity Detectors

Voice Activity Detectors are widely used in current digital age. From simple detector in our favorite communication application to complex detectors used in army to detect suspicious communication. In this paper our aim is to choose the best detector to extract speech from given recording.

Detectors react to change in signal and trigger beginning and end of speech part. This change can be spotted with basic signal volume, its energy, Signal-to-Noise ratio (SNR) and so on. Energy is the most common feature for speech/silence detection. However this feature loses its e□ciency in noisy conditions especially in lower SNRs [4].

There are also other detectors like Mel-Frequency Cepstral Coe□cient detector or Likelihood Ratio Test detector, that are more complex. Google WebRTC detector is used widely on the internet, therefore it was included and looked at.

## 2.1 Short-Term Energy Detector

Energy detector is based on di□erence between total signal energy and short-term energy of each segment.

Recording is sampled and filtered with pre-emphasis. Total energy of the signal is computed as:

$$E_{total} = \frac{1}{N} \cdot \sum^{N-1} s[n]^2, \tag{2.1}$$

where *N* is the number of samples over which energy is calculated. As a percentage value of total energy, $\mathbf{threshold}$ is set.

Threshold is usually between 10 to 20% of total energy, but in special cases another value is set. Then, signal is divided into segments which contains nearly stationary wavelet (Fig. 2.1). Energy of each segment is computed with (Eq. 2.2) and compared with threshold.

$$E_S = \frac{1}{N} \cdot \sum^{N-1} s[n]^2. \tag{2.2}$$

If the short-term energy is bigger than threshold it indicates voiced segment.

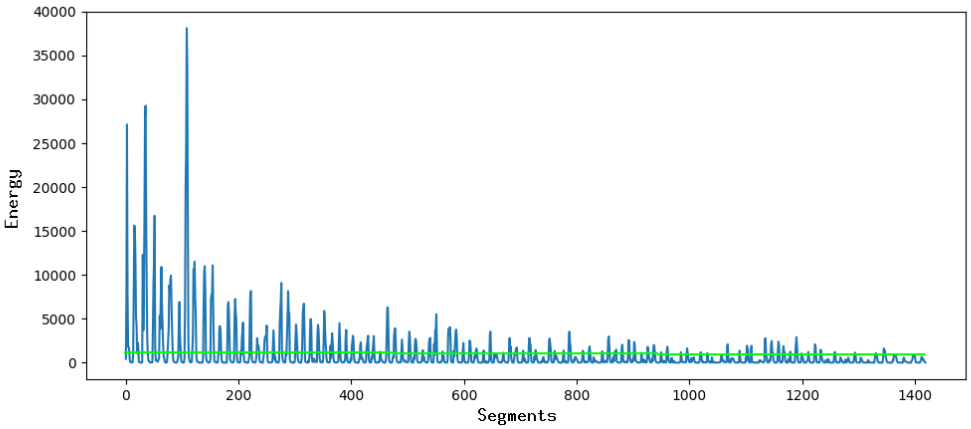


Fig. 2.1: Energy E[S] of each segment with displayed threshold (green)

## 2.2 Volume Detector with Zero-Crossing Rate

Similar to Energy Detector but using absolute values of the signal instead of power. Also, with help of zero-crossing rate algorithm (ZCR) that detects non-speech parts in the recording (see Fig. 2.2). ZCR counts number of changes between positive and negative samples. Algorithm relies on less frequent zero-crossing during the speech and therefore can work with volume detector and improve the results [5].

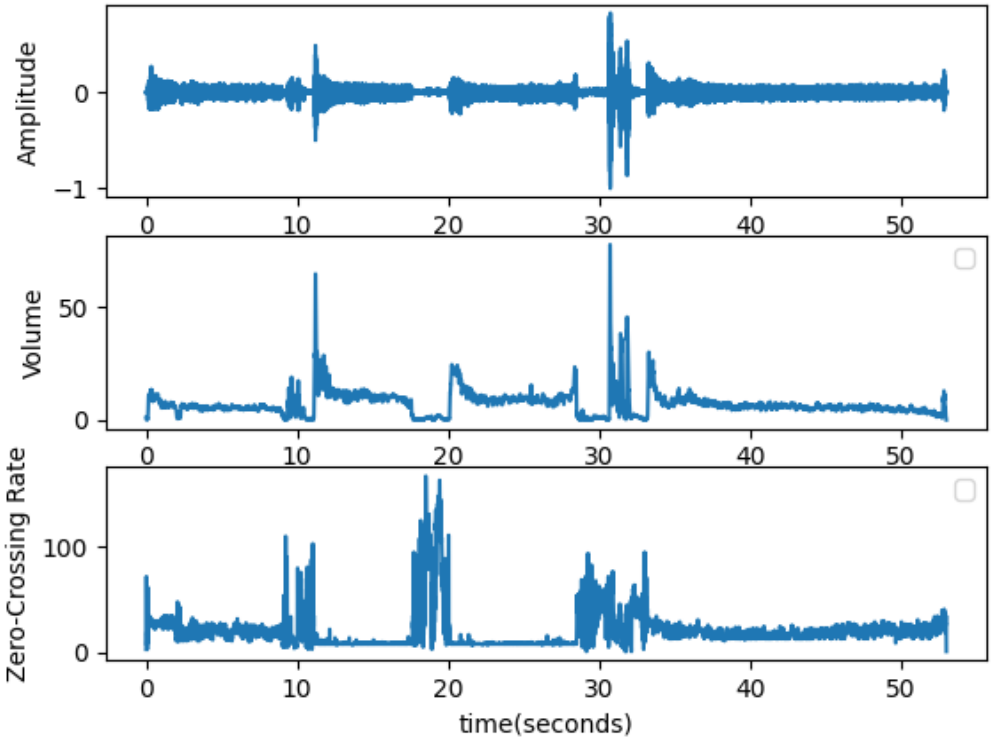


Fig. 2.2: Behavior of volume detector and zero-crossing rate

## 2.3 Likelihood Ratio Test

Statistical likelihood ratio test is a common used voice activity detection method, in which the likelihood ratio of the current frame is compared with adaptive threshold. Threshold is dependent on previous and current frame SNRs. Probability that current frame is speech or noise is computed from geometric mean of individual frequency band likelihood ratios [6].

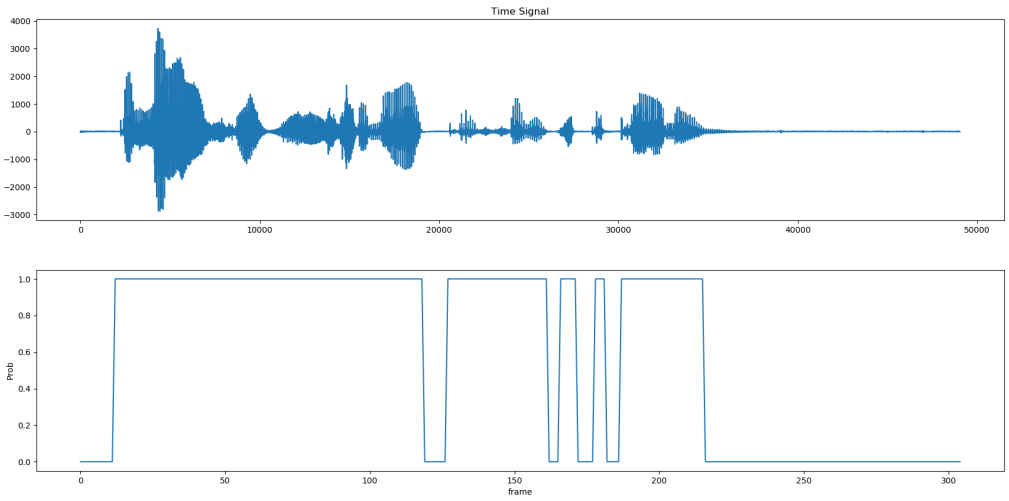


Fig. 2.3: Likelihood ratio performance preview

## 2.4 Mel-Frequency Cepstral Coe□cients

Extracting MFCC features from the signal includes applying pre-emphasis, segmentation, applying STFT (Short-time Fourier transform), taking the log of amplitude at preset Mel frequencies and applying DCT (Discrete Cosine transform).

A Mel is a measure unit based on human hearing. It is approximately linearly spaced bellow 1kHz and logarithmic above. Formula for converting from frequency to Mel scale is:

$$f_{Mel} = 1125.0 \quad \log(1.0 + f/700.0), \tag{2.3}$$

where *f* is a physical frequency in Hz and $f_{Mel}$ is frequency in Mel scale [7].

Pre-emphasis and segmentation is trivial (Ch. 1), after applying STFT to each frame, spectrum is returned. Human ear cannot recognize change in two close frequencies, so taking bunch of similar spectral energy values to sum them up and adding them to one bin. This is performed multiple times, depending on size of Mel filterbank, to cover important frequency range, that is to get an idea how much energy exists in each. This bank consists of narrow triangular filters that widens as the frequency go higher to get less concerned about variations. Human do not hear loudness on a linear scale, so once energy is stored in bins, logarithms of them are taken. At last, beacause bins of energy are overlapping, DCT is applied to bins of energy to decorrelate them. Output of MFCC extraction are coe□cients that indicates energy change on di□erent frequencies. Higher coe□cients change faster with time and therefore can degrade detectors performance, so only first 10 to 15 coe□cients are kept while the others are dropped [8]. In Figure 2.4 there is shown

presence of three speech segments /pa/ – /ta/ – /ka/ visible on zero coe□cients (brighter color means higher coe□cient value).

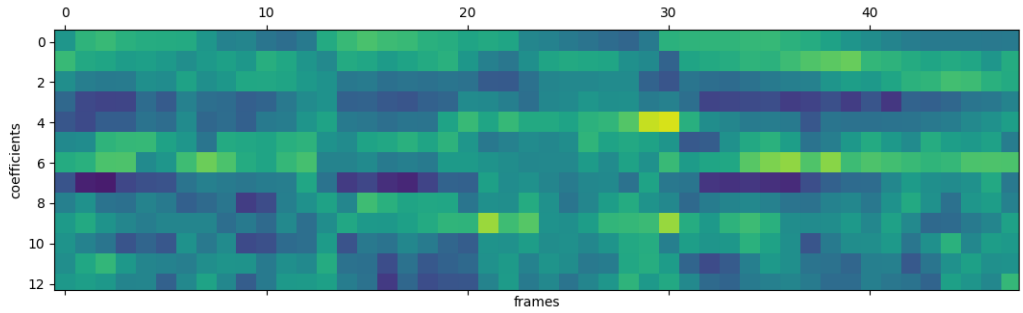


Fig. 2.4: Cepstral coe□cients of syllables /pa/ – /ta/ – /ka/

In order to determine speech from noise Euclidean distance is calculated between Cepstral coe□cients of speech and noise samples. Output is compared with variable threshold value changing with each processed frame by:

```
noise = (0.99 * noise + (1 - 0.99) * frame).
```

Where noise represents mentioned threshold and is a set of MFCC of unvoiced samples with addition of 1% of MFCC of every frame of the processing signal. Basically, each decision whether current frame is a part of speech or not is dependant on previous frames.

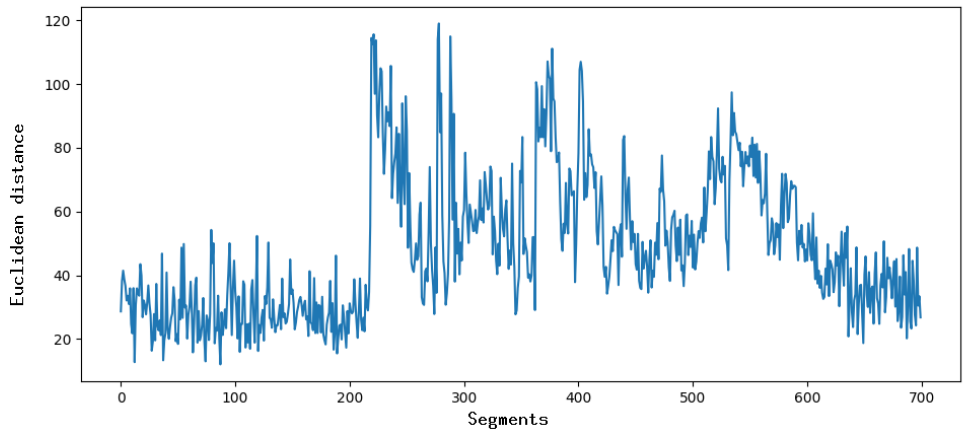


Fig. 2.5: MFCC speech/noise distance preview

## 2.5 Google WebRTC Voice Activity Detection

The VAD that Google developed for the WebRTC project is reportedly one of the best available, being fast, modern and free. It has one *mode* parameter. It represents aggressiveness, which is an integer between 0 and 3. 0 is the least aggressive about filtering out non-speech, 3 is the most aggressive.

First, VAD down-sample input signal to 8kHz to unify process for every recording. Then, uses high-pass filter to remove signal up to 80 Hz. After, it computes logarithms of energy on different frequency intervals (e.g. 80-250Hz) up to 4kHz. With total energy and energies on different intervals it calculates the probabilities for both speech and background noise using Gaussian Mixture Models (GMM). The process combines global likelihood ratio test with local tests for each frequency band. Final decision if frame is voiced is made depending on threshold values set by mode in the beginning. Mode 0 is least aggressive and mode 3 is very aggressive. Each mode has 2 different thresholds (local and global) and everyone of them contains 3 values for different frame length that needs to be set also in the beginning (10, 20 or 30ms). Local threshold sets the frame voice or silence indication while main decision does the global threshold that is compared to summary of logarithms of local likelihood ratios [9].

# 3 Speech Recognition

Speech Recognition is a part of security precautions in modern world. Recognizing and identifying multiple speakers based on properties of a human voice. Voice recognition is used for authentication purposes in more and more applications including property security, banking systems, etc. Voice processing is the successor of image recognition and it provides more options for future development of authentication and verification. Human voice is being recorded throughout the globe almost everywhere. Before 1990, when internet was slowly starting to grow in popularity, there were almost only phone calls being recorded. Few years later when internet and computer technology was presented to the public becoming world wide popular, the number of calls online skyrocketed. Nowadays, especially in current pandemic environment, most of the communications were transferred online. Every call can be recorded with a microphone and voice characteristics can be extracted. For example: frequency, pitch, cadence and tone to determine specific model for each speaker [13].

Speech recognition in this paper is not focused on any authentication or security matter. Recognition of expected words in regards to specific speech task is tested. For example, in task three to task six contained in each recording there are long lasting vowels performed. Vowels should be recognized with predetermined models and compared with suitable algorithm to achieve the best results. Figure 3.1 shows energy of syllables /a/, /i/, /u/ and /a/ in sequence. The visible difference of energy on higher frequencies is the key for syllable recognition.

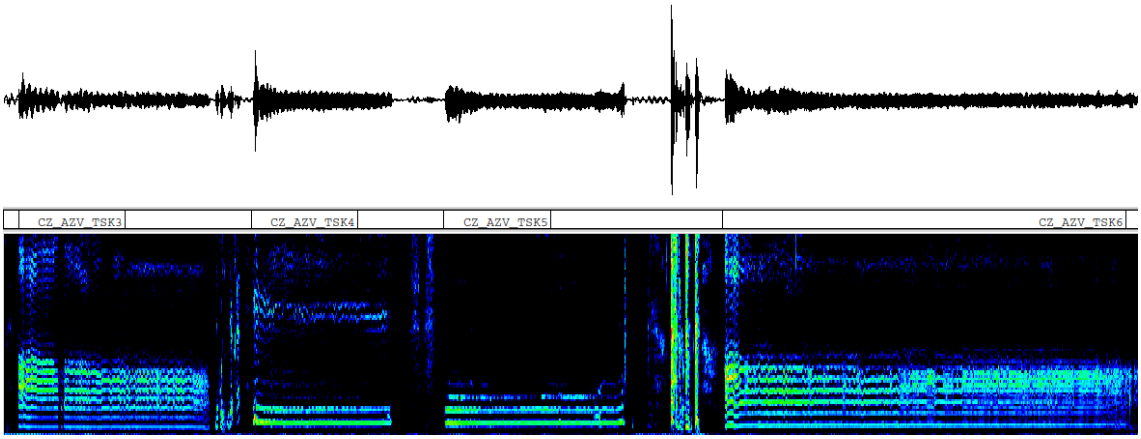


Fig. 3.1: Difference between syllables /a/ /i/ and /u/

## 3.1 Dynamic Time Warping

Proposed algorithm for vowel recognition was dynamic time warping. DTW compares two vectors by expanding or shrinking the time axis of the signal until the match is obtained within two signals. Result of DTW is a distance between two signals with di□erent time series. Algorithm should be able to recognize similar voice spoken in di□erent speeds. It detects similar patterns within voices warping the series non-linearly [14].

### 3.1.1 Using Cepstral coe□cients

Dynamic time warping compares two data series. The results from The performance of recognition systems [14] shows, that using Mel frequency cepstral coe□cients to compare di□erent speech samples is e□cient. Proposed solution is extracting MFC coe□cients from the model and speech sample and calculating euclidean distance between the first 12 coe□cients using DTW. Not all coe□cients are needed as the vowel is sustained for multiple seconds (Fig. 1.1) but the first second is taken only to speed the process.

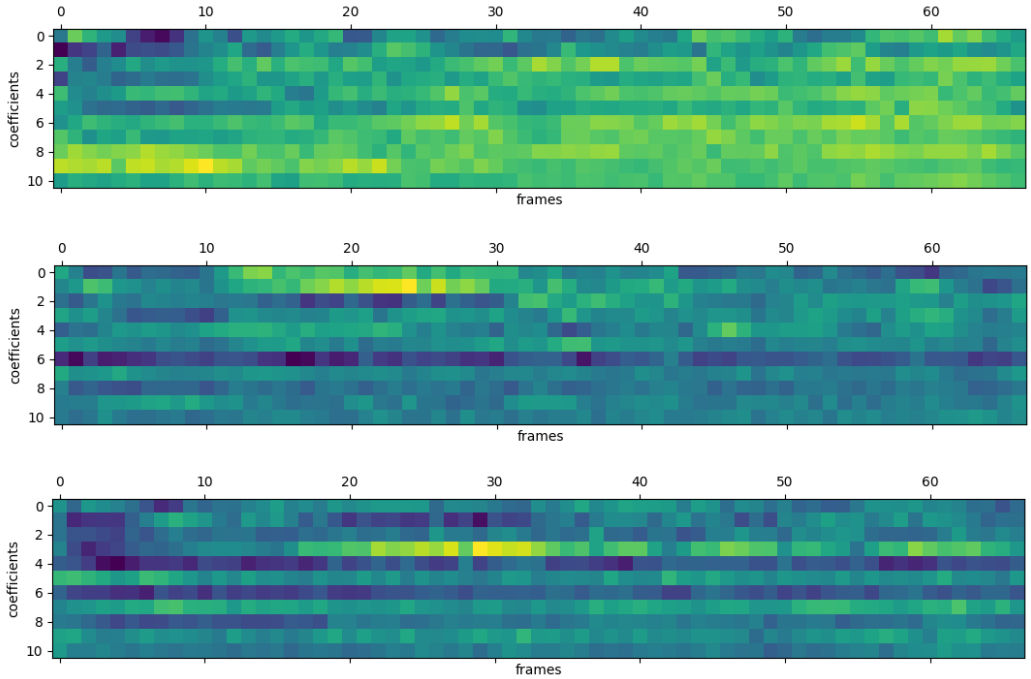


Fig. 3.2: Coe□cients of /a/, /i/ and /u/ vowel

# 4 Database

Speech samples for segmentation tool development were provided by The Brain Diseases Analysis Laboratory. They contain speech tasks, which are used for automatic speech analysis of people with neurological diseases. They are about 5 minutes long and contains up to 17 tasks shown in the table bellow.

Tab. 4.1: Speech tasks

| Label | Speech task | Description |
|---|---|---|
| TSK1 | Monologue | Monolog, at least 90 s long without interruption of a clinician. The participants will be instructed to speak about their hobbies, family, job, actual date activity, etc. |
| TSK2 | Reading | Reading ashorttext.The patient can read the text for her-/himself in advance. |
| TSK3 | Sustained phonation | Approximately 3-s(not longer than 5 s)sustained vowel of /a/ at a comfortable pitch and loudness. Performed on one breath. |
| TSK4 | Sustained phonation | Approximately 3-s (not longer than 5 s)sustained vowel of /i/ at a comfortable pitch and loudness. Performed on one breath. |
| TSK5 | Sustained phonation | Approximately 3-s (not longer than 5 s)sustained vowel of /u/ at a comfortable pitch and loudness. Performed on one breath. |
| TSK6 | Sustained phonation | Sustained phonation of /a/ at a comfortable pitch and loudness as constant and long as possible, at least 5 s. Performed on one breath. |
| TSK7 | Diadochokinetic task | Rapid steady /pa/-/ta/-/ka/ syllables repetition as constant and long as possible, repeated at least 5 times. Performed on one breath. |
| TSK8-17 | Polysyllable word repetition | Repeat 10 polysyllable words according to the clinician. 6 of the words should have at least 3 syllables and CVCV (C –consonant, V –vowel) structure for the first two of them. |

Recordings have 48kHz sampling rate containing 32-bit float values and 768 kbps bit rate resulting in approximately 25MB size. That size is problematic in testing and could take a lot of time. This paper is focused on task detection and for detecting speech, there is no need for 48kHz sampling rate. So, tested recordings were down-sampled to 16kHz with 16-bit depth and bit rate of 256kbps resulting in size of average 10MB.

With the development and testing of VAD detectors on task 7, four recordings were used in Czech language, other four in Hungarian. Half of them have healthy controlled patients, other half patients with Parkinson disease. There is 50% male 50% female samples. Testing of the segmentation tool of the whole recording was tested with six czech recordings.

# 5 Python implementation

Python version 3.8.5 was used for the implementation.

## 5.1 Used libraries

Various libraries were used to manage interface between OS (operating system) file system and gain access to built-in mathematical functions.

For example:

- os, pathlib, glob: managing OS interface (file access)
- numpy, scipy, math: providing mathematical functions (e.g. □t, dct) and reading wave files
- fastdtw: for quick DTW scoring

## 5.2 Voice activity detector structure

Each VAD starts with reading wave file getting sample frequency and signal vector. For example:

```
freqRate,signal=wavfile.read(filePath+fileName+fileExt).
```

Continued with with signal processing for better detector results (Ch. 1.2). From here, detector algorithm is applied. Result is if current frame is voiced or not. If it is, detector sets *VAD* variable to 1 and saves current position in seconds. When it changes back to 0 after several frame cycles, program saves time of beginning from before and current time of triggered *VAD* value change to text file with *.lab* extension as expected result in seconds. This method continues for the whole recording with respect to some constraints. At the end all files are closed and result is stored on preset path with same `fileName` as recording but *.lab* extension for future analysis with wavesurfer application. Example of result text file:

```
0.0000 90.0000 CZ_AZV_TSK1
107.6250 177.5550 CZ_AZV_TSK2
178.9050 183.9050 CZ_AZV_TSK3
189.8000 194.8000 CZ_AZV_TSK4
.
.
282.5100 283.5600 CZ_AZV_TSK15
285.0600 286.1100 CZ_AZV_TSK16
287.7600 288.9600 CZ_AZV_TSK17
```

indicating 17 speech tasks.

With respect to differences between recordings, for example: at start of each recording, unexpected conversation between the recording person and the applicant occurs, unexpected silence lasting a few seconds or other audible sounds. Therefore, first task is cut off as 90 seconds of monologue as stated in the task description (Tab. 4.1), with a room left for improvement in the future.

Detection of beginning of the second task is performed with MFCC detector. Due to faulty task 1, not lasting 90 seconds, expected ending of task 1 is cut off. Therefore, detection of the task 2 starts with ending of task 1. Program removes every frame with higher distance to noise coefficients as noise frames are expected at the end of task 1, shown as red area in Figure 5.1. When the algorithm reaches the silenced part of the recording, grey area in the figure bellow, it switches behavior and now expecting rising edge. Frames are stored in bulks and the distance is averaged. First bulk that exceeds the threshold in distance difference is selected. Then, this bulk and one before is selected as an beginning interval and is closer examined. Precise starting point of the task is determined by bisection of the interval.

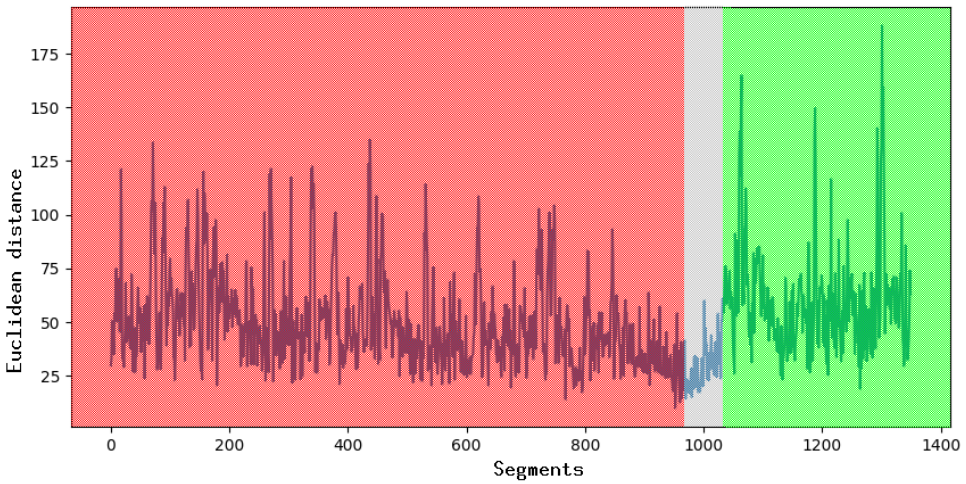


Fig. 5.1: Cut off end of task 1 and task 2 detection

Next tasks are detected similarly with respect to the duration and expected properties of human voice.

## 5.3 Dynamic time warping application

With results from VAD task detection, timestamps for each tasks are available. Use of DTW algorithm in the segmentation tool is proposed to recognize vowels within tasks 3, 4, 5 and 6 containing sustained vowel /a/, /i/ and /u/. The recordings provided for testing do not strictly obey the sequence of tasks. In order to determine which task is performed, DTW o□ers a solution. Creating reference model for each vowel is the most crucial step. Six recordings for each vowel are used and scored with DTW. Recording with best results is saved as a reference model. These three models are used for the recognition purposes.

# 6 Testing

Every task detection is adaptive and can be edited to specific purpose. The precision was tested manually in wavesurfer application. Every recording contains some error, that is disturbing the detectors accuracy.

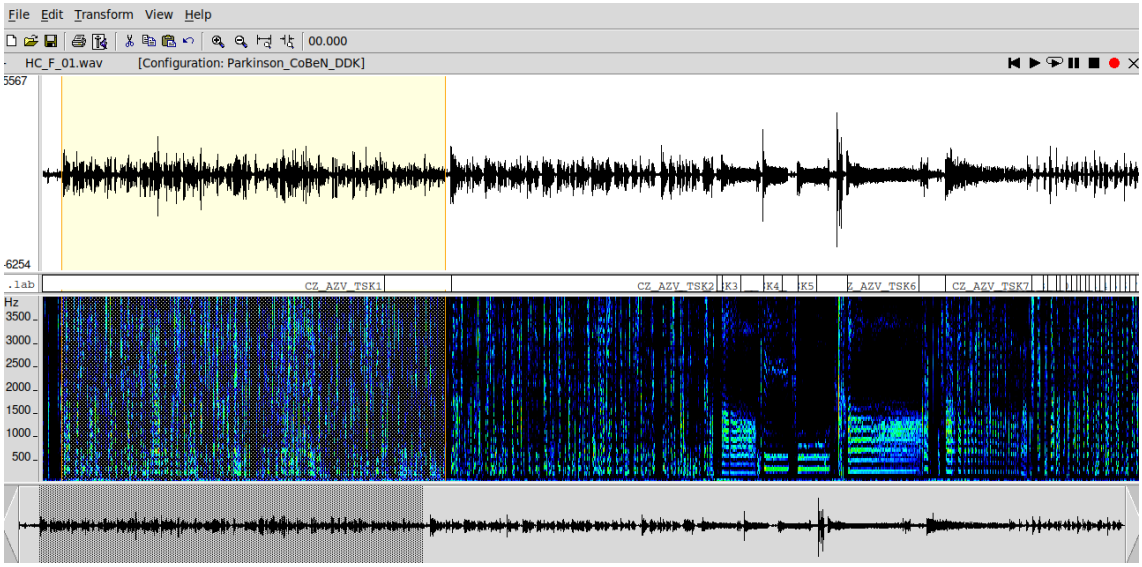


Fig. 6.1: Manual testing in wavesurfer

## 6.1 Segmentation

Reference lab files of the whole recordings were provided from The Brain Diseases Analysis Laboratory. Within the VAD development within task 7, eight samples described in previous chapter were manually labeled using spectrogram in wavesurfer application as prototypes. Wavesurfer interface is show in the picture bellow.

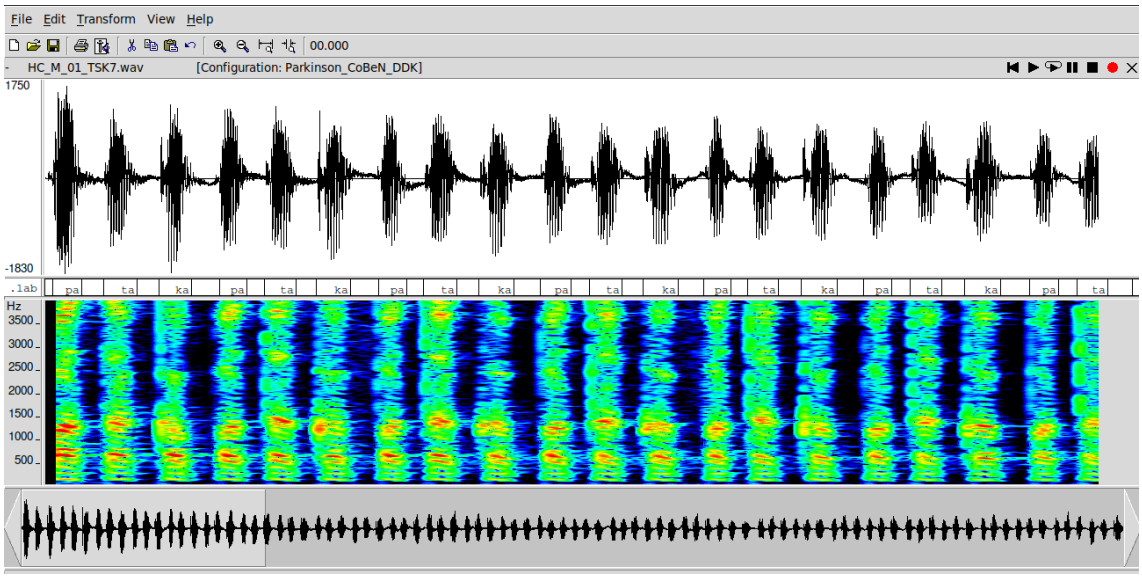


Fig. 6.2: Wavesurfer interface

## 6.2 Receiver Operating Characteristic analysis

Receiver Operating Characteristic – ROC is used as a graphical plot that illustrates the relation between true and false classifier system with variable threshold. Prototypes manually labeled were compared with automatic labels of task 7 during VAD development from detectors with 25ms accuracy. Detectors operating with threshold values were tested throughout the spectrum of available values. For example, Energy VAD with threshold from 0 up to 3 times average energy of the recording was tested. Graph in Figure 6.2 shows how precise is the detector with change of threshold. It displays relation between correct speech detection and false speech detection [11].

Best threshold value with minimal error rate (e.g. threshold: 0.075, error rate: 6.8% for energy VAD) was picked. Then, testing continued with specific error tests counting different relevant errors (Tab. 6.1).

Some detectors do not offer changing threshold as they have built up functions that have only limited use. For example, Google WebRTC VAD offers only 4 levels of speech detection aggressiveness.

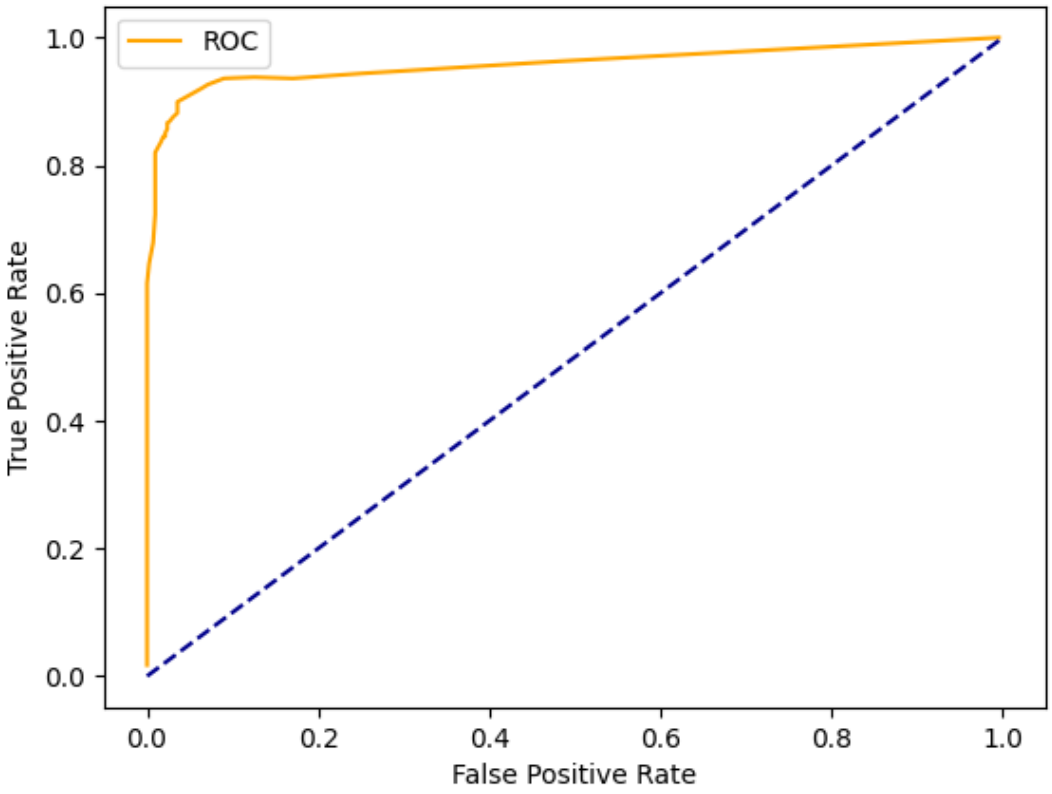


Fig. 6.3: ROC Curve of Energy VAD

## 6.3 Errors in detectors

For further testing, errors are specified:

Tab. 6.1: Error type table

| **Error Type** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| **Activity - 1 Inactivity - 0** | **000** | **001** | **010** | **011** | **100** | **101** | **110** | **111** |
| **VAD Decision** | **010** | **011** | **000** | **001** | **110** | **111** | **100** | **101** |
| **Name** | **NDS** | **FEE** | **WC** | **FEC** | **OVER** | **WB** | **EXT** | **MSC** |

Where 0s and 1s represents unvoiced or voiced frames. Error Description:

- **NDS : Noise detected as speech** — when in the original sample is noised frames but VAD detects speech
- **FEE : Front End Extension** — when VAD detects beginning of the speech before it originaly starts
- **WC : Word Clipping** — when VAD skips original speech frames and assign noise instead
- **FEC : Front End Clipping** — when VAD detects speech later than it originaly started
- **OVER: Prolongated detection of speech in noise**
- **WB : Word Blend** — when VAD blends two speech parts together but originaly they are separated with noise
- **BEC : Back End Clipping** — when VAD ends speech before it originally ended
- **MSC : Midspeech Clipping** — when VAD split one speech into two adding noise in between [12]

# 7 Results

## 7.1 Voice Activity Detectors

Achieved VADs results are shown in tables starting next page. Each showing number and type of error described Table 6.1) as well as total error rate. Total error rate is a count of segments that are false classified (compared to reference segments) divided by total number of segments. Tables are sorted by language and sex starting with Czech speaking healthy male patient following with diseased to di□erentiate with ease. Knowing Hungarian recordings have very low SNR compared with quality of Czech recordings, table (7.1) is showing the best detector for each category. Word Blend error(Tab. 6.1) is picked as most important representing quantity of false identified speech. This is occurring because in TSK7, there is rapid syllables repetition, which means quick change between voiced and unvoiced parts.

Tab. 7.1: Detector quality for Czech and Hungarian recordings

| Recording | Czech (high SNR) | | Hungarian (low SNR) | |
|---|---|---|---|---|
| key variable | average error rate | average WB error | average error rate | average WB error |
| Energy VAD | 10.0 | 0 | 11.4 | 19 |
| Volume VAD | 10.7 | 0 | 11.4 | 13.5 |
| MFCC VAD | 21.7 | 0.5 | 15.9 | 17.8 |
| LRT VAD | 12.2 | 1.3 | 15.6 | 16.3 |
| Google VAD | 37.2 | 2.3 | 19.0 | 18.3 |

Tab. 7.2: Healthy Control CZ Male

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 0 | 7 | 24 | 0 | 14 | 0 | 16 | 0 | 6.8 |
| Volume VAD | 0 | 3 | 40 | 0 | 8 | 0 | 11 | 0 | 6.8 |
| MFCC VAD | 0 | 5 | 41 | 0 | 6 | 0 | 19 | 0 | 7.9 |
| LRT VAD | 3 | 59 | 3 | 0 | 12 | 0 | 29 | 0 | 12.3 |
| Google VAD | 4 | 85 | 2 | 0 | 0 | 0 | 0 | 0 | 44.8 |

Tab. 7.3: Parkinson Disease CZ Male

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 8 | 21 | 42 | 0 | 20 | 2 | 10 | 0 | 12.7 |
| Volume VAD | 6 | 15 | 48 | 0 | 22 | 0 | 9 | 0 | 12.1 |
| MFCC VAD | 0 | 50 | 8 | 0 | 0 | 2 | 85 | 0 | 14.2 |
| LRT VAD | 1 | 38 | 21 | 0 | 23 | 2 | 22 | 0 | 10.1 |
| Google VAD | 0 | 107 | 0 | 0 | 0 | 3 | 1 | 0 | 29.4 |

Tab. 7.4: Healthy Control CZ Female

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 10 | 17 | 39 | 0 | 23 | 0 | 34 | 11 | 14.2 |
| Volume VAD | 29 | 32 | 33 | 1 | 18 | 0 | 27 | 1 | 17.2 |
| MFCC VAD | 0 | 5 | 40 | 50 | 3 | 0 | 55 | 0 | 42.5 |
| LRT VAD | 8 | 68 | 10 | 0 | 24 | 3 | 25 | 2 | 15.1 |
| Google VAD | 57 | 27 | 62 | 3 | 7 | 6 | 9 | 8 | 33.4 |

Tab. 7.5: Parkinson Disease CZ Female

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 1 | 12 | 16 | 0 | 20 | 0 | 10 | 0 | 6.5 |
| Volume VAD | 2 | 23 | 9 | 0 | 18 | 0 | 10 | 0 | 6.8 |
| MFCC VAD | 0 | 30 | 19 | 0 | 0 | 0 | 69 | 0 | 22.0 |
| LRT VAD | 4 | 54 | 1 | 0 | 19 | 0 | 15 | 0 | 11.4 |
| Google VAD | 62 | 18 | 36 | 0 | 1 | 0 | 0 | 0 | 41.1 |

Tab. 7.6: Healthy Control HU Male

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 2 | 11 | 1 | 0 | 1 | 12 | 0 | 1 | 16.9 |
| Volume VAD | 4 | 10 | 0 | 0 | 0 | 12 | 1 | 0 | 17.1 |
| MFCC VAD | 3 | 12 | 1 | 0 | 0 | 12 | 0 | 0 | 18.6 |
| LRT VAD | 1 | 13 | 0 | 0 | 0 | 12 | 0 | 0 | 21.3 |
| Google VAD | 3 | 12 | 0 | 0 | 0 | 12 | 0 | 0 | 19.1 |

Tab. 7.7: Parkinson Disease HU Male

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 19 | 21 | 21 | 17 | 18 | 41 | 19 | 20 | 11.9 |
| Energy VAD | 5 | 17 | 18 | 0 | 12 | 14 | 17 | 2 | 11.9 |
| Volume VAD | 4 | 15 | 16 | 0 | 14 | 14 | 13 | 2 | 9.3 |
| MFCC VAD | 1 | 53 | 0 | 0 | 0 | 19 | 1 | 9 | 26.9 |
| LRT VAD | 6 | 36 | 6 | 0 | 5 | 16 | 10 | 0 | 14.9 |
| Google VAD | 8 | 47 | 3 | 0 | 0 | 19 | 0 | 0 | 21.0 |

Tab. 7.8: Healthy Control HU Female

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 14 | 8 | 25 | 0 | 9 | 11 | 12 | 1 | 7.3 |
| Volume VAD | 5 | 22 | 13 | 0 | 13 | 11 | 21 | 3 | 6.7 |
| MFCC VAD | 18 | 71 | 0 | 0 | 2 | 17 | 0 | 0 | 20.5 |
| LRT VAD | 10 | 35 | 11 | 2 | 9 | 12 | 23 | 2 | 13.0 |
| Google VAD | 20 | 58 | 4 | 0 | 0 | 17 | 0 | 0 | 22.6 |

Tab. 7.9: Parkinson Disease HU Female

| Error Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total err |
|---|---|---|---|---|---|---|---|---|---|
| Name | NDS | FEE | WC | FEC | OVER | WB | EXT | MSC | % |
| Energy VAD | 2 | 12 | 23 | 0 | 12 | 12 | 7 | 9 | 9.4 |
| Volume VAD | 5 | 24 | 16 | 0 | 8 | 17 | 3 | 3 | 10.0 |
| MFCC VAD | 1 | 45 | 2 | 0 | 1 | 23 | 2 | 9 | 15.2 |
| LRT VAD | 1 | 48 | 0 | 0 | 0 | 25 | 0 | 0 | 13.2 |
| Google VAD | 1 | 47 | 0 | 0 | 0 | 25 | 0 | 0 | 13.1 |

## 7.2 Dynamic Time Warping

Matching test was performed with 18 samples, 6 of each syllable /a/, /i/, /u/. Each sample was run through DTW with every other sample and the one with highest match was selected. Only 8 syllables were matched correctly. DTW algorithm worked with 1 second sample of 12 MFC coe□cients.

In the table bellow under testing column are tested samples marked F for female/M for male, followed by number corresponding to the recording from which sample was taken and a/i/u indicating which syllable. Under match column, there is label of matched sample and under distance is the calculated distance by DTW algorithm for best match.

Tab. 7.10: Syllable /a/ /i/ /u/ matching results

| Testing | Match | Distance |
|---|---|---|
| F_1_a | M_1_a | 109.6 |
| F_1_i | F_3_i | 113.2 |
| F_1_u | F_3_a | 99.68 |
| F_2_a | M_3_i | 78.31 |
| F_2_i | M_2_u | 99.09 |
| F_2_u | F_2_a | 124.7 |
| F_3_a | M_2_u | 85.67 |
| F_3_i | F_2_a | 105.8 |
| F_3_u | M_3_i | 91.23 |
| M_1_a | M_2_a | 77.75 |
| M_1_i | M_3_i | 98.49 |
| M_1_u | M_3_i | 108.9 |
| M_2_a | M_1_a | 77.29 |
| M_2_i | M_2_a | 107.9 |
| M_2_u | M_3_u | 79.74 |
| M_3_a | F_2_a | 91.45 |
| M_3_i | F_2_a | 78.77 |
| M_3_u | M_2_u | 79.40 |

Second set of tests was performed between 6 samples of the same syllable. Green columns represent matched result with previous test. This test was performed to show the di□erence between minimal distance of the same syllable and the minimal distance from all syllable samples. For example, F_1_u in Table 7.10 finds best match in F_3_a with distance of 99.68, but in second test (Tab. 7.13) 101.7. Di□erence of only 2 points in the distance resolved in false DTW match.

Tab. 7.11: Tested required distance for syllable /a/ match

| Testing | F_1 | F_2 | F_3 | M_1 | M_2 | M_3 |
|---|---|---|---|---|---|---|
| Match | M_1 | M_3 | M_1 | M_2 | M_1 | F_2 |
| Distance | 109.63 | 93.59 | 97.64 | 77.75 | 77.30 | 91.46 |

Tab. 7.12: Tested required distance for syllable /i/ match

| Testing | F_1 | F_2 | F_3 | M_1 | M_2 | M_3 |
|---|---|---|---|---|---|---|
| Match | F_3 | M_3 | F_1 | M_3 | M_3 | M_1 |
| Distance | 113.26 | 118.82 | 113.29 | 98.49 | 117.04 | 100.36 |

Tab. 7.13: Tested required distance for syllable /u/ match

| Testing | F_1 | F_2 | F_3 | M_1 | M_2 | M_3 |
|---|---|---|---|---|---|---|
| Match | F_3 | M_3 | F_1 | F_3 | M_3 | M_2 |
| Distance | 101.72 | 125.96 | 102.41 | 115.30 | 79.75 | 79.41 |

# 8 Discussion

## 8.1 Voice Activity Detection for task 7

This paper sums up looking for ideal speech detection algorithm for given recordings. Testing achieved 10.0% average error rate and 0 average word blend errors for male and female with Energy VAD for Czech recordings (with low SNR). For Hungarian recordings Energy and Volume detectors have similar results (both with 11.4% error rate), but word blend error occurred less in Volume detector, therefore Volume performed as the best for low SNR recordings (Tab. 7.1). There was no significant difference found in detection between male and female recordings.

MFCC VAD was implemented using only first cepstral coefficient; consequently, performed with significant error.

Tests has shown how world-wide used detectors like Google WebRTC can have problem with specific recording where there is frequently changing speech and silence in the recording. With the error rate 10% higher compared to short-term energy detector. In the future, detectors must be changeable to work with more diverse input. Expectations from Google's detector were higher but it should perform better with more generic speech recordings available for more testing. There is much work to do to analyze behaviour of the detectors on more recordings.

## 8.2 Task detection and dynamic time warping

MFCC detector was updated to work with 12 coefficients and computing euclidean distance between recordings and unvoiced sample. Behavior of reworked detector is very dependant on clarity of the speech. Any speech unrelated to the task is hard to detect and avoid. Also, coughs and clearing throat is significantly impacting the cepstrum of the samples and overall results of the detector. In order to get better results in the future, more strict approach to recording the tasks is needed, as it would significantly improve results of the detector. Pause and resume of the recording between tasks can be a possible solution to avoid undesired input and it could provide enough clarity for more generalized detector.

Application of DTW was used in tasks 3 – 6 to detect and mark correct task in the lab file. Tasks 3 – 6 are not always in sequence by the protocol, so DTW was implemented to detect the task based on MFCC similarity. Results showed only 40% accuracy, therefore not satisfying. More study and testing is needed for application of dynamic time warping. In addition, speech recognition could be used in the future for tasks 7 – 17 to recognize patterns and detect different syllables attached to each task, as the order of the tasks is not strict.

# Conclusion

Voice detection tested for task 7 with low enough error rate is relatively easy to implement for expected input recording. It gets harder for complex recording with more noise, unexpected stops or multiple voices. The goal was to implement speech detector that detects beginning and end of speech parts and saves them in output file for further analysis. The goals were achieved as Energy VAD performed as the best detector for TSK7 with caution of higher word blend error rate for low SNR recordings. Also, Google VAD was tested and compared with detectors for TSK7.

Additionally, detector for the whole recording was built and manually examined and compared to the reference segmentation. Speech recognition using dynamic time warping with mel frequency cepstral coe□cients provided unexpected results as cepstrum of the same syllable had a significant di□erence between speakers.

# Bibliography


[1] SHANNON, C.E., 1949. *Communication in the Presence of Noise.* Proceedings of the IRE, 37(1), pp.10–21. Available at: https://doi.org/10.1109/jrproc.1949.232969.

[2] SMÉKAL, Z. *Zpracování řeči.* Brno: Vysoké učení technické v Brně, 2012. s. 1-171. ISBN: 978-80-214-4896- 4.

[3] DENG, L., O'SHAUGHNESSY, D. *Speech Processing: A Dynamic and Optimization-Oriented Approach* Signal Processing and Communications, CRC Press, 2018. ISBN: 9781482276237

[4] MOATTAR, M. H., HOMAYOUNPOUR, M. M.: *A simple but e cient real-time Voice Activity Detection algorithm*, 2009 17th European Signal Processing Conference, Glasgow, 2009, pp. 2549-2553.

[5] M. Jalil, F. A. Butt and A. Malik, *Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals*, 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2013, pp. 208-212, doi: 10.1109/TAEECE.2013.6557272.

[6] RAMIREZ, J., SEGURA, J. C., BENITEZ, C., GARCIA, L. and RUBIO, A.: *Statistical voice activity detection using a multiple observation likelihood ratio test* in IEEE Signal Processing Letters, vol. 12, no. 10, pp. 689-692, Oct. 2005, doi: 10.1109/LSP.2005.855551.

[7] X. HUANG, X., ACERO, A. and HON, H. *Spoken Language Processing: A guide to theory, algorithm, and system development*, Prentice Hall, 2001.

[8] On, C. K., Pandiyan, P. M., Yaacob, S. and Saudi, A.: *Mel-frequency cepstral coe cient analysis in speech recognition* 2006 International Conference on Computing & Informatics, Kuala Lumpur, 2006, pp. 1-5, doi: 10.1109/ICOCI.2006.5276486.

[9] WISEMAN, J., *Python interface to the WebRTC Voice Activity Detector (VAD)*, Available at: https://github.com/wiseman/py-webrtcvad.

[10] ENQUING, D., GUIZHONG, L., YATONG, Z. and YU, C. *Voice activity detection based on short-time energy and noise spectrum adaptation*,6th International Conference on Signal Processing, 2002., Beijing, China, 2002, pp. 464-467 vol.1.

[11] FAN, J., UPADHYE, S., WORSTER, A. *Understanding receiver operating characteristic (ROC) curves* Canadian Journal of Emergency Medicine 2006, 8(1), pp. 19-20, doi:10.1017/S1481803500013336

[12] ROSCA, J., BALAN, R., FAN, N.P., BWAUGEANT, C. and GILG, V. *Multichannel voice detection in adverse environments*, Proc. ofEUSIPCO, vol. 1, pp. 251–254, Sept. 2002.

[13] B. H. Juang and Tsuhan Chen, *The past, present, and future of speech processing*, in IEEE Signal Processing Magazine, vol. 15, no. 3, pp. 24-48, May 1998, doi: 10.1109/79.671130.

[14] H.MANSOUR, Abdelmajid, Gafar ZEN ALABDEEN SALH and Khalid A. MOHAMMED. *Voice Recognition using Dynamic Time Warping and Mel-Frequency Cepstral Coe cients Algorithms.* International Journal of Computer Applications. 2015, 116(2), 34-41. DOI: 10.5120/20312-2362. ISSN 09758887. Available at: http://research.ijcaonline.org/volume116/number2/pxc3902362.pdf

# Symbols and abbreviations

| | |
|---|---|
| $H(z)$ | Transfer function of a filter |
| $Hz$ | Hertz – frequency unit |
| $\alpha$ | pre-emphasis filter coe□cient |
| $H(\theta)$ | Hamming window function |
| $E_{total}$ | Total energy of the signal |
| $N$ | number of samples |
| $E_S$ | Energy of the current segment |
| $f$ | frequency |
| $f_{Mel}$ | frequency in Mel scale |
| **VAD** | Voice Activity Detector |
| **ZCR** | Zero-Crossing Rate |
| **ROC** | Receiver Operating Characteristic |
| **SNR** | Signal-to-noise ratio |
| **FIR** | Finite Impulse Response |
| **DTFT** | Discrete-Time Fourier Transform |
| **WebRTC** | Real-time communication for the web |
| **MFCC** | Mel-frequency cepstral coe□cients |
| **DCT** | Discrete Cosine transform |
| **GMM** | Gaussian Mixture Model |
| **DTW** | Dynamic Time Warping |
| **OS** | Operating System |
| **TSK7** | Diadochokinetic Task 7 (Tab. 4.1) |
| **Wavesurfer** | WaveSurfer is an open source tool for sound visualization and manipulation |

# A Content of the electronic attachment

Archive contains two folders, one with source codes and one with graphs and images. In source folder there are eleven python files compatible with python version 3.8 and higher. Main executable file is tsk_vad.py containing the tool. File can be executed with python3, taking wave file as an argument. Also, source folder contains five folders, each containing one tested detector.

```
/.....................................................root of the attached archive
├── image ..................................................... graphic files
│   ├── detect.png
│   ├── dtw_res.txt
│   ├── energy_th.png
│   ├── lrl_img.png
│   ├── mfcc.png
│   ├── mfcc_vad.png
│   ├── mfcc_vowels.png
│   ├── preemph.png
│   ├── process.png
│   ├── roc_energy.png
│   ├── segmentation.png
│   ├── testing_wavesurfer.png
│   ├── tsk2_b.png
│   ├── tsk3to6.png
│   ├── tsk_types.png
│   ├── volume_zcr.png
│   └── wavesurfer.png
└── source.....................................................source codes
    ├── Google_VAD
    │   └── WebRTC_VAD.py
    ├── lrt_VAD
    │   ├── LRT_estnoise_ms.py
    │   └── LRT_VAD.py
    ├── MFCC_VAD
    │   ├── mfcc_vad1.py
    │   └── MFCC_VAD.py
    ├── ste_VAD
    │   └── e_vad.py
    ├── volume_VAD
    │   ├── Volume.py
    │   ├── ZeroCR.py
    │   ├── SFE_VAD_t123.py
    │   └── Vol_ZCR_graph_example.py
    ├── concat_s_nons.py
    ├── dtw_test.py
    └── E_VAD.py
```

```
├── features.py
├── GMM_ms.py
├── graph.py
├── lab_score.py
├── proc_wav.py
├── rm_end_labels.py
├── test_code.py
└── tsk_vad.py
```